Symbolic Alternative Characterizations of Testing Preorder for Regular Timed Processes(Concurrency Theory and Applications '96)

AUTHOR(S):

Yuen, Shoji; Sakabe, Toshiki; Inagaki, Yasuyoshi

数理解析研究所講究録
996巻 1997年 5-22

Symbolic Alternative Characterizations of Testing Preorder for Regular Timed Processes

Shoji Yuen Toshiki Sakabe
Yasuyoshi Inagaki
Information Engineering Department,
Nagoya University
{yuen,sakabe,inagaki}@nuie.nagoya-u.ac.jp

Abstract

This paper presents testing preorders for real-time communicating processes with "time-out" as the time constraint primitive where the time domain is the real numbers. The testing preorders presented here are a natural extension of De Nicola and Hennessy's for untimed communicating processes.

We give fully abstract characterizations of the testing preorders in order to ease a proof by focusing the "essential pattern" of tests to establish the preorders. Next, by applying Holmer-Larsen-Wang's "symbolic" method to our preorders, we give the symbolic alternative characterizations where we treat time intervals instead of time itself. For the regular class of real-time, since timeout points are finitely represented, these symbolic characterizations give a proof technique for the testing preorders. Finally, we illustrate our proof technique by some examples.

1 Introduction

Recently the communicating process model has been extended so that it can deal with the "timed" behavior. The communicating process calculi such as CCS[15], CSP[13] and ACP[2] only deal with the temporal behavior of communication where only the pattern of events are treated. Although this abstraction is useful and the calculi have been successfully investigated, it is often important to observe the timed behavior in the practical point of view. For instance, when modeling a communication protocol, the "time-out" behavior is sometimes essential to show for the protocol to work correctly. To fulfill this requirement, many extended calculi of real-time communicating processes have been proposed to deal with the timed behavior [1][4][7][8][11][16][19][20][23].

We regard the time domain as non-negative real numbers and our calculus is extended by a "time-out combinator", that is a binary combinator as an extension of the choice combinator of CCS. As the semantics for the calculus, we define *testing preorders* by extending those for (untimed) CCS by De Nicola and Hennessy[10][18].

We first preset alternative characterizations that are fully abstract to the testing preorders, focusing the essential event patterns in the same manner as for the untimed communicating processes. Alternative characterizations provide very useful proof technique for testing preorders. But these first characterizations are not quite adequate since it requires the infinite checking procedure with respect to time. For this purpose, we next present "symbolic" alternative characterizations that are fully

abstract to the first characterizations by abstracting some finite number of time intervals from the time domain. This abstraction can be finitely possible for the regular class of real-time communicating processes. These second characterizations provide a proof technique for the regular class of real-time communicating processes.

The time-out combinator, written by $\triangleleft_d$, is a binary infix combinator where d is a non-negative real number. A real-time communicating process $P \triangleleft_d Q$ behaves like P if P commits any observable event within time d, and behaves like Q otherwise. Exactly after time d, the behavior is nondeterministically either like P or like Q. Namely, $P \triangleleft_d Q$ makes a choice after time d and is considered as an extension of the choice operator of CCS. This kind of time-out combinator is introduced as the primitive in the literature[8][11] [19][20] and is shown that the combinator can model the variety of timed behavior.

Following the testing principle[18], a real-time communicating process is seen as a kind of black box and known only by interacting with synchronous communications. An examiner of a real-time communicating process, called a "timed test", is also a process with time constraints and some successful terminating states. While testing, the examiner may perform an interaction in an eager way, i.e. an observable event interacts as soon as it becomes available both for the process and for the examiner. We also assume that the passage of time is identical on both sides.

As are done for the untimed processes, we define the testing preorders $\sqsubseteq_{\mathbf{may}}$ and $\sqsubseteq_{\mathbf{must}}$ to understand the timed behavior of the process being examined, where $\sqsubseteq_{\mathbf{may}}$ and $\sqsubseteq_{\mathbf{must}}$ characterize the possibility and the necessity of interaction respectively. In the untimed case, $\sqsubseteq_{\mathbf{may}}$ is alternatively characterized by the inclusion of the languages of events, and $\sqsubseteq_{\mathbf{must}}$ by the possibility and necessity after a sequence of events. Also for the real-time communicating processes, we can give such alternative characterizations where $\sqsubseteq_{\mathbf{may}}$ is the inclusion of the languages of timed events and $\sqsubseteq_{\mathbf{must}}$ by the possibility and necessity accompanied time after a sequence of timed events.

However, when we consider the alternative characterizations as a proof technique for the testing preorder, these characterizations are inadequate. Since we take the time domain as the nonnegative real number, we have to consider the infinitely many timed events to see when the events happen. Again, we consider when are the "essential timing" to characterize the preorders. Focusing on that the essential timing is the time-out point that can be known from the syntax of a real-time communicating process, we will give the finite characterizations for the regular class of real-time communicating processes. This idea for real-time communicating processes is originally presented for bisimulation semantics in [17]. We apply the idea to the testing semantics and develop the "symbolic" alternative characterizations as a proof technique.

The structure of the paper is as follows. In section 2, we define a calculus for timed processes. In section 3, testing preorders are introduced in a standard manner and the first type of alternative characterizations is presented. Section 4 presents a symbolic alternative characterization. In 5, we compare our result with others and present concluding remarks.

2 Timed Process Calculus

Let $\mathbf{R}^+$ be the set of positive real numbers and $\mathbf{R}^{\geq 0} = \mathbf{R}^+ \cup \{0\}$. A set of observable actions is given by Act ranged over by $a, b, a_i, b_i, \ldots$. $\Delta = \{\epsilon(c) \mid c \in \mathbf{R}^+\}$ is defined

as a set of *delay labels* following [23]. We write $\mathcal{A}$ for $Act \cup \Delta$, ranged over by ξ. A set of process variables is given by $\mathcal{V}$ ranged over by $x, y, \ldots$. We assume a constant ∞ that extends $\mathbf{R}^{\geq 0}$, satisfying the following conditions for all $r \in \mathbf{R}^{\geq 0}$ with respect to the arithmetic operations:

- $r < \infty$
- $r + \infty = \infty + r = \infty$
- $\infty - r = \infty$

And we write $\mathbf{R}^{\infty}$ for $\mathbf{R}^{\geq 0} \cup \{\infty\}$. A class of *timed process expressions* is defined by the following BNF:

$$E ::= \mathbf{nil} \mid a.E \mid E_1 + E_2 \mid E_1 \oplus E_2 \mid E_1 \lhd_d E_2 \mid x \mid \mathbf{rec}{:}x.E$$

where $a \in Act$, $d \in \mathbf{R}^{\geq 0}$ and $x \in \mathcal{V}$. We write $\mathcal{PE}$ for the set of timed process expressions. All occurrences of x in the context $\mathbf{rec}:x.E$ are *bound*, and *free* otherwise. For $E, F \in \mathcal{PE}$, $E[F/x]$ denotes the timed process expression where all free occurrences of x in E are simultaneously replaced by F. A timed process expression is *closed* if it contains no free occurrence of process variables. Following [17], we require the *action guardedness condition* for a timed process expression, that is defined as follows:

Definition 1 *A process variable x is* guarded *in E if all occurrences of x appear in the form of $a.F$ which is a subexpression of E. We say E is guarded when all the process variables appearing in E are guarded in E and x is guarded in F where every subexpression of E in the form $\mathbf{rec}:x.F$.*

As a basic operational semantics, we define a transition system $\langle \mathcal{PE}, \rightarrow, \mathcal{A} \rangle$ with the *derivation relation* $\rightarrow \subseteq (\mathcal{PE} \times \mathcal{PE}) \cup (\mathcal{PE} \times \mathcal{A} \times \mathcal{PE})$ shown in table 1. We denote $E \rightarrow E'$ if $(E, E') \in \rightarrow$ and $E \xrightarrow{\xi} E'$ if $(E, \xi, E') \in \rightarrow$.

E is called a *regular timed process* if it is closed and guarded. The set of regular timed processes is denoted by $\mathcal{P}$.

From the construction of the SOS rules and the definition of $\mathcal{P}$, we get the following closure property of the closed timed process.

Lemma 1 *If $P \in \mathcal{P}$, then $\{E \mid P \rightarrow E\} \cup \{E \mid P \xrightarrow{\xi} E, \xi \in \mathcal{A}\} \subseteq \mathcal{P}$.*

Henceforth, we focus on the sub-LTS $\langle \mathcal{P}, \rightarrow, \mathcal{A} \rangle$ to exactly model the timed processes. For this sub-LTS, we use the following notation[1]:

- $P \rightarrow$ if for some P' $P \rightarrow P'$.
- $P \xrightarrow{\xi}$ if for some P' $P \xrightarrow{\xi} P'$.
- $P \rightarrow_0 P'$ if $P (\rightarrow)^* P'$.
- $P \rightarrow_c P'$ if $P (\rightarrow)^* \circ \xrightarrow{\epsilon(c)} \circ (\rightarrow)^* P'$.
- $P \Rightarrow_d P'$ if $P \xrightarrow{c_1} \circ \cdots \circ \xrightarrow{c_n} P'$ for $c_i \in \mathbf{R}^{\geq 0}$ where $1 \leq i \leq n$ and $d = \Sigma_{i=1}^{n} c_i$.
- $P \xrightarrow{a}_d P'$ if $P \Rightarrow_d \circ \xrightarrow{a} P'$.

Inaction	$\dfrac{}{\mathbf{nil} \xrightarrow{\epsilon(c)} \mathbf{nil}}$
Prefix	$\dfrac{}{a.E \xrightarrow{a} E}$ $\qquad$ $\dfrac{}{a.E \xrightarrow{\epsilon(c)} a.E}$
Time-out	$\dfrac{E \xrightarrow{a} E'}{E \triangleleft_d F \xrightarrow{a} E'}$ $\qquad$ $\dfrac{E \longrightarrow E'}{E \triangleleft_d F \longrightarrow E' \triangleleft_d F}$ $\dfrac{E \xrightarrow{\epsilon(c)} E'}{E \triangleleft_d F \xrightarrow{\epsilon(c)} E' \triangleleft_{d-c} F}$ if $d \geq c$ $\qquad$ $\dfrac{}{E \triangleleft_0 F \longrightarrow F}$
External Choice	$\dfrac{E \longrightarrow E'}{E + F \longrightarrow E' + F}$ $\qquad$ $\dfrac{F \longrightarrow F'}{E + F \longrightarrow E + F'}$ $\dfrac{E \xrightarrow{a} E'}{E + F \xrightarrow{a} E'}$ $\qquad$ $\dfrac{E \xrightarrow{a} E'}{E + F \xrightarrow{a} F'}$ $\dfrac{E \xrightarrow{\epsilon(c)} E' \ , \ F \xrightarrow{\epsilon(c)} F'}{E + F \xrightarrow{\epsilon(c)} E' + F'}$
Internal Choice	$\dfrac{}{E \oplus F \longrightarrow E}$ $\qquad$ $\dfrac{}{E \oplus F \longrightarrow F}$
Recursion	$\dfrac{}{\mathbf{rec}{:}x.E \longrightarrow E[\mathbf{rec}{:}x.E/x]}$

Table 1: Operational Semantics by derivations

- $S(P) = \{a \in Act \mid P(\rightarrow)^* \circ \stackrel{a}{\rightarrow}\}$.

The derivation relation have the following basic properties ([7][17][23]):

Lemma 2 *1. (transition liveness) $P \not\rightarrow$ or $P \stackrel{\epsilon(c)}{\longrightarrow}$ for some $c \in \mathbf{R}^+$.*

2. (time determinacy) $P \stackrel{\epsilon(c)}{\longrightarrow} P'$ and $P \stackrel{\epsilon(c)}{\longrightarrow} P''$ imply $P' \equiv P''$.

3. (time continuity) For $c_1, c_2 \in \mathbf{R}^+$, $P \Rightarrow_{c_1+c_2} P'$ if and only if $P \Rightarrow_{c_1} P''$ and $P'' \Rightarrow_{c_2} P'$ for some P''.

4. (maximal progress) $P \rightarrow$ implies $P \stackrel{\epsilon(c)}{\not\longrightarrow}$.

5. (strong persistency) Whenever $P \stackrel{\epsilon(c)}{\longrightarrow} P'$ for some $c \in \mathbf{R}^+$, $P \stackrel{a}{\rightarrow}$ if and only if $P' \stackrel{a}{\rightarrow}$.

Proof: Induction on the structure of P. □
Note that the strong persistency property holds because the time-out operator performs the empty unobservable transition. Thus, if we take the weak transition, the property does not hold as in[16].

3 Testing Preorders

3.1 Timed tests

We define the class of *timed tests* as the examiner by the following BNF:

$$T ::= \mathbf{nil} \mid a.T \mid w.T \mid T_1 \lhd_d T_2 \mid T_1 + T_2 \mid T_1 \oplus T_2$$

where $a \in Act$, $d \in \mathbf{R}^{\geq 0}$ and $w \notin Act$. The set of all timed tests is denoted by $\mathcal{T}$. The operational semantics for the timed tests is given by a transition system $\langle \mathcal{T}, \rightarrow, \mathcal{A} \cup \{w\} \rangle$ where $\rightarrow$ is defined by table 1 in addition to the following SOS rule:

$$\frac{}{w.T \stackrel{w}{\longrightarrow} T}$$

A timed test is essentially same as a timed process besides that it has no recursion power and there is a special action prefix w to indicate the successful termination of interaction. This means that a process is observed by a finite sequence of observable actions.

3.2 Interaction System

The interaction system is a composition of a timed process and a timed test. The interaction between a timed process and a timed test is formally defined as follows:

Definition 2 *An* interaction system *for $\mathcal{P}$ and $\mathcal{T}$, written as $\mathcal{I}(\mathcal{P}, \mathcal{T})$, is a transition system $\langle \mathcal{I}, \rightarrow_I \rangle$ where the states $\mathcal{I} = \mathcal{P} \times \mathcal{T}$ and the transition relation $\rightarrow_I \subseteq \mathcal{I} \times \mathbf{R}^{\geq 0} \times \mathcal{I}$ defined by table 2. Following the literature [10], [18], we write $P \| T$ for a state $\langle P, T \rangle \in \mathcal{I}$. We write $P \| T \stackrel{c}{\rightarrow}_I P' \| T'$ when $\langle P \| T, c, P' \| T' \rangle \in \rightarrow_I$.*

$$\frac{P \xrightarrow{\epsilon(c)} P' \ , \ T \xrightarrow{\epsilon(c)} T'}{P\|T \xrightarrow{c}_I P'\|T'} \text{ if } S(P) \cap S(T) = \emptyset$$

$$\frac{P \longrightarrow P'}{P\|T \xrightarrow{0}_I P'\|T} \qquad \frac{T \longrightarrow T'}{P\|T \xrightarrow{0}_I P\|T'}$$

$$\frac{P \xrightarrow{a} P' \ , \ T \xrightarrow{a} T'}{P\|T \xrightarrow{0}_I P'\|T'} \text{ where } a \in Act$$

Table 2: Interaction System

Definition 3 *For an interaction system $\mathcal{I}(\mathcal{P}, \mathcal{T})$, a state $P\|T$ is* successful *whenever $T \xrightarrow{w}$. A (possibly infinite) sequence of transitions:*

$$P_0\|T_0 \xrightarrow{c_1}_I P_1\|T_1 \xrightarrow{c_2}_I \cdots \xrightarrow{c_n}_I P_n\|T_n \xrightarrow{c_{n+1}}_I \cdots$$

is a computation from $P_0\|T_0$ *if one of the following conditions is satisfied:*

(1) The sequence is finite and the last state is successful

(2) The sequence is infinite and $\Sigma_i c_i$ is infinite.

A computation satisfying (1) is called successful.

Note that different from the untimed case, the dead-locked failure does not happen due to the transition liveness property. We exclude the "Zeno-phenomenon" by condition (2).

We now establish testing preorders as usual.

Definition 4 *Let $P \in \mathcal{P}$ and $T \in \mathcal{T}$.*

1. *P* **may** *T if some computation from $P\|T$ is successful.*
2. *P* **must** *T if every computation from $P\|T$ is successful.*

Definition 5 *Let $\mathcal{E} \subseteq \mathcal{T}$.*

1. *$P \sqsubseteq^{\mathcal{E}}_{\mathbf{may}} Q$ if for all $T \in \mathcal{E}$ P* **may** *T implies Q* **may** *T.*
2. *$P \sqsubseteq^{\mathcal{E}}_{\mathbf{must}} Q$ if for all $T \in \mathcal{E}$ P* **must** *T implies Q* **must** *T.*
3. *$P \sqsubseteq^{\mathcal{E}} Q$ if $P \sqsubseteq^{\mathcal{E}}_{\mathbf{may}} Q$ and $P \sqsubseteq^{\mathcal{E}}_{\mathbf{must}} Q$.*

When $\mathcal{E} = \mathcal{T}$, we simply write $P \sqsubseteq_{\mathbf{may}} Q$, $P \sqsubseteq_{\mathbf{must}} Q$ and $P \sqsubseteq Q$.

[1] If necessary, we also use this notation for $\langle \mathcal{PE}, \rightarrow, \mathcal{A} \rangle$.

3.3 Alternative Characterizations

Here we give the alternative characterizations for the testing preorders defined above. The characterizations are the direct extensions of those for the untimed communicating processes by reflecting the "essential" pattern of testing.

A finite sequence of pairs $\langle d_i, a_i \rangle$ where $d_i \in \mathbf{R}^+$ and $a_i \in Act$:

$$\langle d_1, a_1 \rangle \langle d_2, a_2 \rangle \cdots \langle d_n, a_n \rangle$$

is called a *timed word* and a set of timed words is called a *timed language*. The empty timed word is denoted by λ and the set of all timed word is denoted by $\mathcal{L}_{Act} = (\mathbf{R}^{\geq 0} \times Act)^*$. When $P \stackrel{a_1}{\rightarrow}_{d_1} \circ \stackrel{a_2}{\rightarrow}_{d_2} \cdots \circ \stackrel{a_n}{\rightarrow}_{d_n} P'$, we write $P \stackrel{\sigma}{\Rightarrow} P'$ for a timed word $\sigma = \langle a_1, d_1 \rangle \langle a_2, d_2 \rangle \cdots \langle a_n, d_n \rangle$.

First we show that $\sqsubseteq_{\mathbf{may}}$ is characterized by the inclusion relation between timed languages.

Definition 6 *Let $P \in \mathcal{P}$. The* timed language of P *is defined by* $L(P) = \{\sigma \mid P \stackrel{\sigma}{\Rightarrow}\}$.

For a timed word σ, a timed test $T_0(\sigma)$ is recursively defined with respect to σ as follows:

$$\begin{aligned} T_0(\lambda) &= w.\mathbf{nil} \\ T_0(\langle d, a \rangle \sigma') &= \mathbf{nil} \lhd_d (a.T_0(\sigma') \lhd_0 \mathbf{nil}) \end{aligned}$$

We write $\mathcal{T}_0 = \{T_0(\sigma) \mid \sigma \in \mathcal{L}_{Act}\}$.

Lemma 3 *$P \sqsubseteq^{\mathcal{T}_0}_{\mathbf{may}} Q$ if and only if $P \sqsubseteq_{\mathbf{may}} Q$*

Proof: The only-if part is obvious since $\mathcal{T}_0 \subseteq \mathcal{T}$. For if part, suppose P **may** T. Then, there exists a successful computation from $P \| T$. For the computation, there exists a transition of T such that $T \stackrel{\sigma}{\Rightarrow} T'' \Rightarrow_d T' \stackrel{w}{\rightarrow}$. Thus, P **may** $T_0(\sigma)$. By the assumption, Q **may** $T_0(\sigma)$. Then, $Q \| T \stackrel{c_1}{\rightarrow}_I \cdots \stackrel{c_n}{\rightarrow}_I Q' \| T''$ and by the transition liveness property (lemma 2(1)) Q **may** T. □

Lemma 4 *$\sigma \in L(P)$ if and only if P **may** $T_0(\sigma)$.*

Theorem 1 *$P \sqsubseteq_{\mathbf{may}} Q$ if and only if $L(P) \subseteq L(Q)$.*

Proof: The theorem follows from lemma 3 and 4. □

Next we show an alternative characterization of the must preorder.

Definition 7 *$P \ll_{\mathbf{must}} Q$ if the following holds:*

> *For $\sigma \in L(Q)$ and $c \in \mathbf{R}^{\geq 0}$, $Q \stackrel{\sigma}{\Rightarrow} \circ \Rightarrow_c Q'$ implies $S(P') \subseteq S(Q')$ for some P' such that $P \stackrel{\sigma}{\Rightarrow} \circ \Rightarrow_c P'$.*

Let $\gamma \in \mathbf{R}^+$, $\sigma \in \mathcal{L}_{Act}$ and $a \in Act$. For a finite subset of Act, $A = \{a_1, \ldots, a_n\}$ and $d \in \mathbf{R}^{\geq 0}$, timed tests $T_1^\gamma(\sigma, a, d)$ and $T_2^\gamma(\sigma, A, d)$ are recursively defined as follows:

$$\begin{aligned} T_1^\gamma(\lambda, a, d) &= \mathbf{nil} \lhd_d (a.\mathbf{nil} \lhd_\gamma w.\mathbf{nil}) \\ T_1^\gamma(\langle d_1, a_1 \rangle \sigma, a, d) &= \mathbf{nil} \lhd_{d_1} (a_1.T_1^\gamma(\sigma, a, d) \lhd_\gamma w.\mathbf{nil}) \\ T_2^\gamma(\lambda, A, d) &= \mathbf{nil} \lhd_d ((a_1.w.\mathbf{nil} + \cdots + a_n.w.\mathbf{nil}) \lhd_\gamma \mathbf{nil}) \\ & \qquad A = \{a_1, a_2, \ldots, a_n\} \\ T_2^\gamma(\langle d_1, a_1 \rangle \sigma, A, d) &= \mathbf{nil} \lhd_{d_1} (a_1.T_2^\gamma(\sigma, A, d) \lhd_\gamma \mathbf{nil}) \end{aligned}$$

We write $\mathcal{T}_e(\gamma) = \{T_1^\gamma(\sigma, a, d), T_2^\gamma(\sigma, A, d) \mid \sigma \in \mathcal{L}_{Act}, finite A \subseteq Act, a \in Act, d \in \mathbf{R}^{\geq 0}\}$.

We shall show $\mathcal{T}_e(\gamma)$ is essential to verify $\sqsubseteq_{\mathbf{must}}$ if γ is small enough to distinguish processes.

Lemma 5 *For $\gamma \in \mathbf{R}^+$, $P \sqsubseteq_{\mathbf{must}}^{\mathcal{T}_e(\gamma)} Q$ implies $L(Q) \subseteq L(P)$.*

Proof: By induction on the length of σ, it is shown that P **must** $T_1^\gamma(\sigma, a, d)$ if and only if $\sigma\langle d, a\rangle \in L(P)$.

Let $\sigma = \sigma'\langle d, a\rangle \in L(Q)$. Then, for all $\gamma \in \mathbf{R}^+$, Q **must** $T_1^\gamma(\sigma, a, d)$. By the assumption P **must** $T_1^\gamma(\sigma, a, d)$. Thus, $\sigma \in L(P)$. □

Lemma 6 *For $\gamma \in \mathbf{R}^+$, $P \sqsubseteq_{\mathbf{must}}^{\mathcal{T}_e(\gamma)} Q$ implies $P \ll_{\mathbf{must}} Q$.*

Proof: Suppose for all $\gamma \in \mathbf{R}^+$, $P \sqsubseteq_{\mathbf{must}}^{\mathcal{T}_e(\gamma)} Q$ and $P \not\ll_{\mathbf{must}} Q$. Let $Q \stackrel{\sigma}{\Rightarrow} \circ \Rightarrow_d Q'$, then $P \stackrel{\sigma}{\Rightarrow} \circ \Rightarrow_d P'$ for some P' such that $S(P') \not\subseteq S(Q')$.

Take a such that $a \in S(P')$ and $a \notin S(Q')$. Then, either $Q' \stackrel{a}{\not\rightarrow}_0$ or $Q' \stackrel{a}{\rightarrow}_c$ for some $c \in \mathbf{R}^+$. For the former case, for all $\gamma \in \mathbf{R}^+$, $T_2^\gamma(\sigma, \{a\}, d)$ and P **must** $T_2^\gamma(\sigma, \{a\}, d)$. For the latter case, take the least c' for such c. Then, for $\gamma < c'$, Q **must** $T_2^\gamma(\sigma, \{a\}, d)$ and P **must** $T_2^\gamma(\sigma, \{a\}, d)$. This complete the proof since a contradiction is derived for both cases. □

Lemma 7 *$P \ll_{\mathbf{must}} Q$ implies $P \sqsubseteq_{\mathbf{must}} Q$.*

Proof: Suppose $P \ll_{\mathbf{must}} Q$ and we show that Q **must** T implies P **must** T for all $T \in \mathcal{T}$. Let Q **must** T, then there exists an infinite computation from $Q\|T$ such that $Q\|T \stackrel{c_1}{\rightarrow}_I Q_1\|T_1 \stackrel{c_2}{\rightarrow}_I \cdots \stackrel{c_n}{\rightarrow}_I Q_n\|T_n \stackrel{c_{n+1}}{\rightarrow}_I \cdots$ where n is such that for all $m > n$ $S(T_{m-1}) = S(T_m)$, $S(Q_m) \cap S(T_m) = \emptyset$ and $T_n \stackrel{w}{\not\rightarrow}$. Such n exists since a timed test is finite. Thus, for some $\sigma \in \mathcal{L}_{Act}$ and d, $Q \stackrel{\sigma}{\Rightarrow} \circ \Rightarrow_d Q_n$. By the definition of $\ll_{\mathbf{must}}$, there exists P_k such that $P \stackrel{\sigma}{\Rightarrow} \circ \Rightarrow_d P_k$ and $S(P_k) \subseteq S(Q_n)$. Thus, there exists a sequence of transition: $P\|T \stackrel{c_1'}{\rightarrow}_I \cdots \stackrel{c_k'}{\rightarrow}_I P_k\|T_n$. And T_n will not change in the following transitions since $S(P_k) \subseteq S(Q_n)$. This concludes P **must** T. □

Theorem 2 *$P \sqsubseteq_{\mathbf{must}} Q$ if and only if $P \ll_{\mathbf{must}} Q$.*

Proof: For "only-if" part, $P \sqsubseteq_{\mathbf{must}} Q$ implies $P \sqsubseteq_{\mathbf{must}}^{\mathcal{T}_e(\gamma)} Q$ for all γ since $\mathcal{T}_e(\gamma) \subseteq \mathcal{T}$. By lemma 6, $P \ll_{\mathbf{must}} Q$. "if" part is lemma 7. □

From theorem 1, theorem 2 and lemma 5, we obtain the following characterization for the testing preorder.

Corollary 1 *$P \sqsubseteq Q$ if and only if $L(P) = L(Q)$ and $P \ll_{\mathbf{must}} Q$.*

For the alternative characterizations shown above, we have the following useful lemma to prove the preorders for regular timed processes.

Lemma 8 *Let* **rec**$:x.E, P \in \mathcal{P}$. *Then,*

1. *$L(E[P/x]) \subseteq L(P)$ implies $L($**rec**$:x.E) \subseteq L(P)$*
2. *$E[P/x] \ll_{\mathbf{must}} P$ implies* **rec**$:x.E \ll_{\mathbf{must}} P$

$M(x)$	$=$	0	$M(\mathbf{nil})$	$=$	∞
$M(a.E)$	$=$	∞	$M(E_1 \lhd_d E_2)$	$=$	$min(M(E_1), d)$
$M(E_1 + E_2)$	$=$	$min(M(E_1), M(E_2))$	$M(E_1 \oplus E_2)$	$=$	0
$M(\mathbf{rec}{:}x.E)$	$=$	$M(E)$			

Table 3: Life Function M

4 Symbolic Alternative Characterizations

This section gives "symbolic" alternative characterizations as a proof technique. Due to the dense property of a time domain, the alternative characterizations in the previous section still require an infinite checking procedure with respect to time. But there are only finite number of points at which a real-time communicating process may change its behavior through the timeout combinator by the construction of a process. For example, the characterization of $a.\mathbf{nil} \lhd_5 b.\mathbf{nil}$ can be divided into the three cases: before time 5, exactly at time 5 and after time 5. Based on this observation, we will define characterizations with respect to time intervals that are "essential" to observe the behavior of a process.

4.1 Symbolic Transition Relation

We define the *life time function* $M : \mathcal{PE} \to \mathbf{R}^\infty$ by table 3. We also define a *symbolic delay relation* $\overset{d}{\rightsquigarrow} \subseteq \mathcal{P} \times \mathcal{P}$ as follows:

- $P \overset{0}{\rightsquigarrow} P'$ if $P(\rightarrow)^* P'$
- When $M(P) = d < \infty$, $P \overset{\epsilon(d)}{\rightarrow} P''$ and $P'' \overset{d'}{\rightsquigarrow} P'$ imply $P \overset{d+d'}{\rightsquigarrow} P'$

If $P \overset{d}{\rightsquigarrow} P'$ for some P', then we simply write $P \overset{d}{\rightsquigarrow}$.

Lemma 9 *When $P \in \mathcal{P}$, $\{d \mid P \overset{d}{\rightsquigarrow}\}$ is finite.*

Proof: Suppose $\{d \mid P \overset{d}{\rightsquigarrow}\}$ is infinite. By the construction of $\overset{d}{\rightsquigarrow}$, there must be an infinite derivation from P only by $\rightarrow$ and $\overset{\epsilon(c)}{\rightarrow}$. But this is impossible because of the guardedness condition and the definition of computation. □

In the rest of this subsection, we present the basic properties with respect to the labeled transition system $\langle \mathcal{P}, \rightarrow, \mathcal{A} \rangle$.

Lemma 10 *$M(P) = d$ and $0 < d < \infty$ imply $P \overset{\epsilon(d)}{\rightarrow}$.*

Lemma 11 *$P \rightarrow$ implies $M(P) = 0$.*

Lemma 12 *When $P \overset{\epsilon(c)}{\rightarrow} P'$, one of the followings holds:*

1. *$M(P) = \infty$*
2. *$P \overset{c}{\rightsquigarrow} P'$ and $M(P') = 0$*

3. *There exists* d *such that* $P \overset{d}{\rightsquigarrow} P'$, $M(P') > 0$ *and* $d \leq c < d + M(P)$

A *timed action* is a triple $\langle d, a, t \rangle \in \mathbf{R}^{\geq 0} \times Act \times \mathbf{R}^{\infty}$ to express that action a is enabled after time d for time t. We write a_d^t for a timed action $\langle d, a, t \rangle$. The set of all timed actions is denoted by $\mathcal{A}_T$, ranged over by μ, ν. A relation $\preceq$ over $\mathcal{A}_T$, $a_d^t \preceq a_e^u$, holds if $e \leq d$ and $d + t \leq u + e$. $a_d^t \preceq a_e^u$ is said that a_d^t *covers* a_e^u, meaning that action a in a_d^t is possible whenever action a is enabled in a_e^u. We write $a_d^t \sim a_e^u$ when $d < e + u$ and $e < d + t$, meaning that action a may be enabled both in a_d^t and a_e^u at the same time.

A sequence of timed actions is called a *symbolic timed word.* We use the notation for a symbolic timed word. The empty symbolic timed word is denoted by λ_s. And the set of all symbolic timed words is denoted by $\mathcal{L}_T$.

- Let symbolic timed words $\sigma_s = \mu_1 \mu_2 \cdots \mu_n$ and $\sigma_s' = \mu_1' \mu_2' \cdots \mu_n'$. $\sigma_s \sim \sigma_s'$ if $\mu_i \sim \mu_i'$ for $1 \leq i \leq n$.
- Let $\Sigma_s \subseteq \mathcal{L}_T$.

$$\begin{aligned} \Sigma_s \uparrow \lambda_s &= \{\sigma_s \in \Sigma_s \mid |\sigma_s| = 1\} \\ \Sigma_s \uparrow \mu'\sigma' &= \Sigma_s'(\mu') \uparrow \sigma_s' \end{aligned}$$

  where $\Sigma_s'(\mu') = \{\sigma_s | \mu\sigma_s \in \Sigma_s, \mu' \preceq \mu\}$.

- Let $A_T \subseteq \mathcal{A}_T$. A_T **covers** a_d^t if there exists $a_{d_i}^{t_i} \in A_T$ such that $d_1 \leq d$, $d + t \leq d_n + t_n$, $d_i \leq d_{i+1} \leq d_i + t_i$ for $1 \leq i < n$.

Definition 8 *A* symbolic transition relation $\overset{a_d^t}{\longmapsto}$ *is defined as follows:*

$P \overset{a_d^t}{\longmapsto} P'$ *if for some* P'' $P \overset{d}{\rightsquigarrow} P''$, $P'' \overset{a}{\rightarrow} P'$ *and* $M(P'') = t$

Let $\sigma_s = \mu_1 \mu_2 \cdots \mu_n \in \mathcal{A}_T$. *We write* $P \overset{\sigma_s}{\Longmapsto} P'$ *if* $P \overset{\mu_1}{\longmapsto} \circ \cdots \circ \overset{\mu_n}{\longmapsto} P'$.

Intuitively, $P \overset{a_d^t}{\rightarrow} P'$ means that P waits for time d and do an action a before t to become P'. This notion is justified by the time-determinacy property and the time continuity property.

Lemma 13 $P \overset{a}{\rightarrow} P'$ *and* $M(P) > 0$ *imply* $P \overset{a}{\rightarrow}_d P'$ *for all* d *such that* $0 \leq d < M(P)$.

Lemma 14 *Let* $P \overset{a_d^t}{\longmapsto} P'$.

1. $t = 0$ *implies* $P \overset{a}{\rightarrow}_d P'$
2. $t > 0$ *implies* $P \overset{a}{\rightarrow}_d P'$ *for all* d' *such that* $d \leq d' < d + t$

The derivation by the symbolic transition relation is finite.

Lemma 15 *Let* $P \in \mathcal{P}$. $\{\langle P', \mu \rangle \mid P \overset{\mu}{\longmapsto} P', \mu \in \mathcal{A}_T\}$ *is finite.*

Proof: From lemma 9, d's for $\mu = a_d^t \in \{\mu \mid P \overset{\mu}{\longmapsto}\}$ are finitely many. And by the construction of $M(P)$, t's must appear in a timeout operator $\triangleleft_t$ or ∞. And P is finite branching with respect to communication. □

4.2 Symbolic Characterization of $\sqsubseteq_{\mathbf{may}}$

Definition 9 *The* symbolic timed language *of P is defined as follows:*

$$L_s(P) = \{\sigma_s \mid P \stackrel{\sigma_s}{\models\Rightarrow}\}$$

A symbolic timed word specifies a timed language by the timeout points. For example, $L_s(a.\mathbf{nil} \lhd_5 b.\mathbf{nil}) = \{\lambda_s, a_0^5, b_5^\infty\}$ which specifies the timed language $\{\lambda, \langle a,t\rangle, \langle b,u\rangle \mid 0 \le t \le 5, u \ge 5\}$.

Definition 10 *The* covering relation $\trianglelefteq$ *is defined as follows:*

(1) $\lambda_s \in \Sigma_s$ *implies* $\lambda_s \trianglelefteq \Sigma_s$

(2) $(\Sigma_s \uparrow \sigma_s)$ **covers** μ *implies* $\sigma_s\mu \trianglelefteq \Sigma_s$

We write $\Sigma'_s \trianglelefteq \Sigma_s$ *if for all* $\sigma_s \in \Sigma'_s$ $\sigma_s \trianglelefteq \Sigma_s$.

Intuitively $L_s(P) \trianglelefteq L_s(Q)$ means that the timed language specified by $L_s(P)$ is covered by some timed language specified by $L_s(Q)$.

In the rest of this subsection, we shall show that the *covering relation* characterizes $\sqsubseteq_{\mathbf{may}}$. First we show the *adequacy* of the characterization.

Before showing the adequacy result, we show the following technical lemma.

Lemma 16 *Let* $P \stackrel{a}{\rightarrow}_d P'$ *and* $\{Q_1, \ldots, Q_n\} \subseteq \mathcal{P}$, *then,* $L_s(P) \trianglelefteq \cup_i L_s(Q_i)$ *implies* $L_s(P') \trianglelefteq \cup_i \{Q' \mid Q_i \stackrel{a}{\rightarrow}_d Q'\}$.

Proof: Let P'' such that $P \Rightarrow_d P'' \stackrel{a}{\rightarrow} P'$. By repeated applications of Lemma 11 and Lemma 12, either of the following (1) or (2) holds:

(1): $M(P'') = 0$ and $d = d'$

(2): $M(P'') > 0$ and $d' \le d < d' + M(P'')$

We prove for each cases:

(1): $M(P'') = 0$ and $d = d'$

- If $\lambda_s \in L_s(P')$:
  Since $a_t^0 \in L_s(P)$, $L_s(P) \trianglelefteq \cup_{k=1}^n L_s(Q_k)$. Following from this, $(\cup_{k=1}^n \{L_s(Q_k)\} \trianglelefteq \cup_{k=1}^n)$ **covers** a_t^0. Thus, there exist $a_{d_l}^{t_l}$ for $1 \le l \le m$ such that $\{a_{d_1}^{t_1}, \ldots, a_{d_m}^{t_m}\}$ and $d_1 \le d \le d_m + t_m$ where $d_j \le d_{j+1}$ and $d_{j+1} \le d_j + t_j$ $(1 \le l < n)$. Then, for some j, $Q_j \stackrel{a_d^t}{\models\Rightarrow} Q'_j$ and $d_j \le d < d_j + t_j$. Since by lemma 14 $Q_j \stackrel{a}{\rightarrow}_d Q'_j$, $\cup_{k=1}^n \{L_s(Q') | Q_j \stackrel{a}{\rightarrow}_d Q'_j\} \neq \emptyset$. Then, by the definition, $\lambda_s \trianglelefteq \cup_{k=1}^n \{L_s(Q') | Q_k \stackrel{a}{\rightarrow}_d Q'\}$ since $\lambda_s \in L_s(P)$ for every P.
- If $\sigma'_s\mu \in L_s(P')$:
  Since $a_d^0\sigma'_s\mu \in L_s(P)$, $\cup_{k=1}^n L_s(Q_k) \uparrow a_d^0\sigma'_s\mu \in L_s(P)$ **covers** μ. Thus, for $\sigma'_s\mu$ there exists $\mathbf{Q}(\sigma'_s\mu)$ such that $\mathbf{Q}(\sigma'_s\mu) \subseteq \cup_{k=1}^n \{Q' | Q_k \stackrel{\nu}{\longmapsto} Q', a_d^0 \preceq \nu\}$ and $\cup\{L_s(R) | R \in \mathbf{Q}(\sigma'_s\mu)\} \uparrow \sigma'_s$ **covers** μ. Thus, $\sigma'_s\mu \trianglelefteq \mathbf{Q}(\sigma'_s\mu)$. Then since by lemma 14 $\mathbf{Q}(\sigma'_s\mu) \subseteq \cup_{k=1}^n \{Q' | Q_k \stackrel{a}{\rightarrow}_d Q'\}$, $\sigma'_s\mu \trianglelefteq \cup_{k=1}^n \{Q' | Q_k \stackrel{a}{\rightarrow}_d Q'\}$.

Thus, $L_s(P') \trianglelefteq \cup_{k=1}^{n}\{Q'|Q_k \xrightarrow{a}_d Q'\}$ for case (1).

(2): $M(P'') > 0$ and $d' \leq d < d' + M(P'')$

- If $\lambda_s \in L_s(P')$:
  Since $a_{d'}^{M(P'')} \in L_s(P)$, there exist $a_{d_l}^{t_l}$ for $1 \leq l \leq m$ such that $d_1 \leq d'$, $d'+M(P'') \leq d_m$, $d_i \leq d_{i+1}$, $d_{i+1} \leq d_i+t_i$ and $\{a_{d_1}^{t_1}, \ldots, a_{d_m}^{t_m}\} \subseteq \cup_{k=1}^{n} L_s(Q_k)$.
  Since $d' \leq d < d' + M(P')$, for d, there exists j such that $a_{d_j}^{t_j} \sim a_{d'}^{t}$, $d_j \leq d < d_j + t_j$ and $Q_j \overset{a_{d_j}^{t_j}}{\longmapsto} Q'_j$. Since by lemma 14 $Q_j \xrightarrow{a}_d Q'_j$, $\lambda_s \in \cup_{k=1}^{n}\{L_s(Q')|Q_j \xrightarrow{a}_d Q'\}$ showing $\lambda_s \trianglelefteq \{L_s(Q')|Q_j \xrightarrow{a}_d Q'\}$.
- If $\sigma'_s\mu \in L_s(P')$:
  Since $a_{d'}^{M(P'')}\sigma'_s\mu \in L_s(P)$,
  $\cup_{k=1}^{n}\{L_s(R)|Q_k \overset{\nu}{\longmapsto} R, a_{d'}^{M(P'')} \preceq \nu\} \uparrow \sigma'_s$ **covers** μ.
  While, $a_{d'}^{M(P'')} \preceq \nu$ implies that $Q_k \xrightarrow{a}_d R$ when $Q_k \overset{\nu}{\longmapsto} R$ by lemma 14. This establishes $\cup_{k=1}^{n}\{L_s(R)|Q_k \xrightarrow{a}_d R\} \uparrow \sigma'_s$ **covers** μ.
  Thus, $\sigma'_s\mu \trianglelefteq \cup_{k=1}^{n}\{L_s(Q')|Q_k \xrightarrow{a}_d Q'\}$.

Thus, $L_s(P') \trianglelefteq \cup_{k=1}^{n}\{Q'|Q_k \xrightarrow{a}_d Q'\}$ for case (2).

□

Lemma 17 *$L_s(P) \trianglelefteq L_s(Q)$ implies $L(P) \subseteq L(Q)$.*

Proof: We shall prove the following by induction on the length of σ.

For $\sigma \in L(P)$, $L_s(P) \trianglelefteq \cup_{i=1}^{n} L_s(Q_i)$ implies $\sigma \in \cup_{i=1}^{n} L_s(Q_i)$.

For $\sigma = \lambda$, it is trivial. Let $\sigma = \langle d, a \rangle \sigma'$. Then, there exists P' such that $P \xrightarrow{a}_d P'$ and $\sigma' \in L(P')$. By lemma 16, $L_s(P') \trianglelefteq \cup_i\{Q'_{i,j}|Q_i \xrightarrow{a}_d Q'_{i,j}\}$ By the induction hypothesis, $\sigma' \in \cup_{i,j} L(Q'_{i,j})$. Thus, $\sigma \in \cup_i L(Q_i)$.

Taking $\cup_i Q_i$ as $\{Q\}$ completes the proof. □

Next we show the *abstractness* of the characterization. We introduce a function that expands a symbolic timed word to the timed language being specified.

$$
\begin{aligned}
exp(\lambda_s) &= \{\langle d, a \rangle \sigma \mid \sigma \in exp(\sigma_s)\} \\
exp(a_d^0) &= \{\langle d, a \rangle \sigma \mid \sigma \in exp(\sigma_s)\} \\
exp(a_d^t) &= \{\langle d', a \rangle \sigma \mid \sigma \in exp(\sigma_s), d \leq d' < d + t\} \quad (t > 0)
\end{aligned}
$$

Lemma 18 *$L(P) \subseteq L(Q)$ implies $L_s(P) \trianglelefteq L_s(Q)$.*

Proof: We prove the contra-positive. Suppose $L_s(P) \not\trianglelefteq L_s(Q)$. Then there exists $\sigma_s a_d^t \in L_s(P)$ such that $(L_s(Q) \uparrow \sigma_s)$ **coyers** a_d^t.

(1): $L_s(Q) \uparrow \sigma_s = \emptyset$: for $\sigma \in exp(\sigma_s a_d^t)$, $\sigma \in L(P)$ and $\sigma \notin L(Q)$. Namely $L(P) \not\subseteq L(Q)$.

(2): $L_s(Q) \uparrow \sigma_s = \{a_{d_1}^{t_1}, \ldots, a_{d_n}^{t_n}\}$: Since $(L_s(Q) \uparrow \sigma_s)$ covers a_d^t, there exists e such that $d \leq e < t + d$ and $e < d_i$ or $d_i + t_i \leq e$ for all $1 \leq i \leq n$. Then, for $\sigma \in exp(\sigma_s)$, $\sigma\langle e, a\rangle \in L(P)$ and $\sigma\langle e, a\rangle \notin L(Q)$. Thus, $L(P) \not\subseteq L(Q)$.

□

Combination of Lemma 17 with lemma 18 shows the full abstractness.

Theorem 3 *$L_s(P) \trianglelefteq L_s(Q)$ if and only if $L(P) \subseteq L(Q)$.*

Corollary 2 *$L_s(P) \trianglelefteq L_s(Q)$ if and only if $L(P) \sqsubseteq_{\mathbf{may}} L(Q)$.*

Corollary 3 *$L_s(E[P/x]) \trianglelefteq L_s(P)$ implies $L_s(\mathbf{rec}{:}x.E) \trianglelefteq L_s(P)$.*

Thus, together with lemma 15, we can finitely prove $\sqsubseteq_{\mathbf{may}}$ for a regular class of real-time communicating processes.

4.3 Symbolic Characterization of $\sqsubseteq_{\mathbf{must}}$

In this subsection, we present a symbolic alternative characterization for $\sqsubseteq_{\mathbf{must}}$. The underlying idea for the characterization is to consider the essential type of timed tests, shown in definition 7 only for the timeout points.

In the characterization, we use the following "past" notation.

Let $P/\langle \sigma'_s, c\rangle = \{R \mid P \stackrel{\sigma'}{\Longrightarrow} \circ \stackrel{c}{\rightsquigarrow} R, M(R) = 0\}$

$\cup\{R \mid P \stackrel{\sigma'}{\Longrightarrow} \circ \stackrel{c'}{\rightsquigarrow} R, c' \leq c < c' + M(R) \text{ for some } c'\}$.

$P/\langle \sigma'_s, c\rangle$ denotes the set of timed processes after timed word σ'_s is performed time c.

Definition 11 *$P \ll^{\mathsf{sym}}_{\mathbf{must}} Q$ if*
for $\sigma_s \in L_s(Q)$ and $d \in \mathbf{R}^{\geq 0}$ $Q \stackrel{\sigma_s}{\Longrightarrow} \circ \stackrel{d}{\rightsquigarrow} Q'$ implies the following (1) and (2):

(1) $\sigma_s \trianglelefteq \{\sigma'_s \in L_s(P) | \sigma'_s \sim \sigma_s\}$

(2) For $\sigma'_s \sim \sigma_s$ and $\sigma'_s \in L_s(P)$, either of (2-a) or (2-b) holds:

(2-a) When $M(Q') = 0$:
There exists P' such that $P' \in P/\langle \sigma'_s, d\rangle$ and $S(P') \subseteq S(Q')$

(2-b) When $M(Q') > 0$:
For all $c' \in \{d\} \cup \{c | d \leq c < d + M(Q'), P \stackrel{\sigma'_s}{\Longrightarrow} \circ \stackrel{c}{\rightsquigarrow}\}$, there exists P' such that $P' \in P/\langle \sigma'_s, c'\rangle$ and $S(P') \subseteq S(Q')$.

We shall show the equivalence between $\ll^{\mathsf{sym}}_{\mathbf{must}}$ and $\ll_{\mathbf{must}}$ in the rest of this subsection. First we show the adequacy of the characterization.

Lemma 19 *$P \ll^{\mathsf{sym}}_{\mathbf{must}} Q$ implies $P \ll_{\mathbf{must}} Q$.*

Proof: Suppose $Q \stackrel{\sigma}{\Rightarrow} Q'' \Rightarrow_d Q'$ and $P \ll^{\mathsf{sym}}_{\mathbf{must}} Q$. We will show the existence of P' and P'' such that $P \stackrel{\sigma}{\Rightarrow} P'' \Rightarrow_d P'$ and $S(P') \subseteq S(Q')$.

Let $\sigma = \langle d_1, a_1\rangle \cdots \langle d_n, a_n\rangle$, then there exist some symbolic word $\sigma_s = \mu_1 \cdots \mu_n$ such that for each $\mu_i = a_{e_i}^{u_i}$ either $u_i = 0$ and $d_i = e_i$ or $u_i > 0$ and $e_i \leq d_i < e_i + u_i$. Then, $Q \stackrel{\sigma_s}{\Longrightarrow} Q''$. And also $\sigma \in exp(\sigma_s)$.

For $Q''\Rightarrow_d Q'$, let $Q''\overset{d_s}{\rightsquigarrow}Q'$ then either $M(Q')=0$ and $d_s=d$ or $M(Q')>0$ and $d_s\leq d<d_s+M(Q')$ hold.

From condition (1) of definition 11 and lemma 17, there exists σ'_s such that $\sigma\in expand(\sigma'_s)$ and $\sigma'_s\sim\sigma_s$. For this σ'_s, we will show either of condition (2) holds.

- If $M(Q')=0$:

  When $P'\in\{R|P\overset{\sigma'_s}{\Longrightarrow}\circ\overset{d}{\rightsquigarrow}R, M(R)=0\}$ and $S(P')\subseteq S(Q')$, $P\overset{\sigma'_s}{\Longrightarrow}\circ\Rightarrow_d P'$. From the fact that $\sigma\in exp(\sigma'_s)$, $P\overset{\sigma}{\Rightarrow}\circ\Rightarrow_d P'$.

  Otherwise, $P'\in\{R|P\overset{\sigma'_s}{\Longrightarrow}\circ\overset{d}{\rightsquigarrow}R, M(R)>0, c'\leq d<c'+M(R)\}$ and $S(P')\subseteq S(Q')$. $P\overset{\sigma'_s}{\Longrightarrow}P''\overset{d}{\rightsquigarrow}P'$. Since $\sigma\in exp(\sigma'_s)$ and $P''\Rightarrow_d P'$ by lemma 12(3), $P\overset{\sigma}{\Rightarrow}P''\Rightarrow_d P'$.

- If $M(Q')>0$:

  Let $C=\{c|d_s\leq c<d_s+M(Q'), P\overset{\sigma'_s}{\Longrightarrow}\circ\overset{c}{\rightsquigarrow}\}$. If $C=\emptyset$, then there exists P' such that $P'/\langle\sigma'_s,d_s\rangle$ and $S(P')\subseteq S(Q')$. Let $P\overset{\sigma'_s}{\Longrightarrow}P''\overset{e}{\rightsquigarrow}P''$, then $d_s+M(Q')\leq e+M(P')$. Then, for all e' such that $d_s\leq e'<d_s+M(Q')$, $P''\Rightarrow_{e'}P'$.

  If $C=\{c_i|i=1,2,\cdots\}$, for all e, $c_1=\leq e<d_s+M(Q')$ implies that there exists i such that $c_i=e$ and $c_i\leq e<d_s+M(Q')$. Let R_j such that $P\overset{\sigma'_s}{\Longrightarrow}\circ\overset{c_j}{\rightsquigarrow}R_j$, then by the assumption $S(R_j)\subseteq S(Q')$ for all j. Then, if $c_1\leq d<d_s+M(Q')$, then there exists i such that $P\overset{\sigma'_s}{\Longrightarrow}\circ\rightarrow_d P'=R_i$. If $d_s\leq d<c_1$, there exists some $P'\in P/\langle\sigma'_s,d_s\rangle$ such that $P\overset{\sigma'_s}{\Longrightarrow}\circ\rightarrow_d P'$.

□

Next we present the abstractness of the characterization.

Lemma 20 *$P\ll_{\mathbf{must}}Q$ implies $P\ll^{\mathsf{sym}}_{\mathbf{must}}Q$.*

Proof: We derive a contradiction supposing $P\ll_{\mathbf{must}}Q$ and $P\not\ll^{\mathsf{sym}}_{\mathbf{must}}Q$.

Let $Q\overset{\sigma_s}{\Longrightarrow}\circ\overset{d_s}{\rightsquigarrow}Q'$. We also suppose $\Sigma'_s=\{\sigma'_s\in L_s(P)|\sigma_s\sim\sigma'_s\}$ and $\sigma_s\trianglelefteq\Sigma'_s$. Then, there must be $\rho_s\in L_s(P)$ such that $\rho_s\sim\sigma_s$ and none of condition (2-a) and (2-b) holds.

- If $M(Q')=0$:
  For all $P'\in P/\langle\rho_s,d_s\rangle$, $S(P')\not\subseteq S(Q')$. Then, for some $\sigma\in exp(\rho_s)$ and $d=d_s$, $\{R|P\overset{\sigma}{\Rightarrow}\circ\rightarrow_d R\}\subseteq P/\langle\rho_s,d_s\rangle\}$. Then, for $Q\overset{\sigma}{\Rightarrow}\circ\Rightarrow_d Q'$, there is no P' such that $P\overset{\sigma}{\Rightarrow}\circ\Rightarrow_d P'$ and $S(P')\subseteq S(Q')$. This contradicts $P\ll_{\mathbf{must}}Q$.

- If $M(Q')>0$:
  Let $\sigma\in exp(\sigma_s)\cap exp(\sigma'_s)\neq\emptyset$. Then, from lemma 12(3), for d such that $d_s\leq d<d_s+M(Q'')$, $Q\overset{\sigma}{\Rightarrow}Q'\rightarrow_d Q'$. From $P\ll_{\mathbf{must}}Q$, for all such d, there must exist P' such that $P\overset{\sigma}{\Rightarrow}\circ\rightarrow_d P'$ and $S(P')\subseteq S(Q')$. But this makes condition (2-b) established. Then, this contradicts $P\not\ll^{\mathsf{sym}}_{\mathbf{must}}Q$.

□

From lemma 19 and lemma 20, we obtain the following characterization.

Theorem 4 *$P\ll^{\mathsf{sym}}_{\mathbf{must}}Q$ if and only if $P\ll_{\mathbf{must}}Q$.*

Corollary 4 $P \ll^{\mathsf{sym}}_{\mathbf{must}} Q$ *if and only if* $P \sqsubseteq_{\mathbf{must}} Q$.

Corollary 5 $E[P/x] \ll^{\mathsf{sym}}_{\mathbf{must}} P$ *implies* $\mathbf{rec}{:}x.E \ll^{\mathsf{sym}}_{\mathbf{must}} P$.

Together with lemma 15, we can finitely check the preorder for the regular class of real-time communicating processes.

4.4 Examples

We end this section by showing some examples of proofs using the symbolic alternative characterizations.

For notational convenience, we simply write a for $a.\mathbf{nil}$ in what follows.

Example 1 *Let* $P_2 = (\mathbf{nil} \lhd_5 a) \oplus (\mathbf{nil} \lhd_6 b)$ *and* $Q_2 = \mathbf{nil} \lhd_5 (a \lhd_1 (a + b))$. *Then,* $L_s(P_2) = \{\lambda_s, a_5^\infty, b_6^\infty\}$ *and* $L_s(Q_2) = \{\lambda_s, a_5^1, a_6^\infty, b_6^\infty\}$. *Since* $L_s(P_2) \trianglelefteq L_s(Q_2)$ *and* $L_s(Q_2) \trianglelefteq L_s(P_2)$, $L_s(P_2) \sqsubseteq_{\mathbf{may}} L_s(Q_2)$ *and* $L_s(Q_2) \sqsubseteq_{\mathbf{may}} L_s(P_2)$.

Thus, for P_2 *and* Q_2, *condition (1) of definition 11 holds.*

For $Q_2 \overset{5}{\rightsquigarrow} (a \lhd_1 b)$, $S(a \lhd_1 b) = \{a\}$ *and* $M(a \lhd_1 b) = 1$. *Then,* $P_2 \overset{5}{\rightsquigarrow} a$ *and* $M(a) = \infty$ *make condition (2-b) hold.*

For $Q \overset{6}{\rightsquigarrow} a \lhd_0 (a + b)$, $P_2 \overset{5}{\rightsquigarrow} a$ *satisfies condition (2-a).*

For $Q_2 \overset{6}{\rightsquigarrow} a + b$, $P_2 \overset{6}{\rightsquigarrow} b$ *satisfies condition (2-b). And for* $Q_2 \overset{a_5^1}{\Longrightarrow} \mathbf{nil}$, $Q_2 \overset{a_6^\infty}{\Longrightarrow} \mathbf{nil}$. *For* $P_2 \overset{a_5^\infty}{\Longrightarrow} \mathbf{nil}$ *and* $Q_2 \overset{b_6^\infty}{\Longrightarrow} \mathbf{nil}$, $P_2 \overset{b_6^\infty}{\Longrightarrow} \mathbf{nil}$ *Thus,* $P_2 \ll^{\mathsf{sym}}_{\mathbf{must}} Q_2$.

But for $P_2 \overset{\lambda_s}{\Longrightarrow} \circ \overset{6}{\rightsquigarrow} b$, $\{c | 6 \leq c, Q_2 \overset{\lambda_s}{\Rightarrow} \circ \overset{c}{\rightsquigarrow}\} = \{6\}$. *And* $P/\langle \lambda_s, 6 \rangle = \{a, a + b\}$. *Then, there is no symbolic transition that satisfies condition (2-b). Therefore,* $Q_2 \not\ll^{\mathsf{sym}}_{\mathbf{must}} P_2$.

Example 2 *For the vending machine example[17], we can show* P_3 *that is less deterministic than* Q_3 *with respect to timeout is related less in the sense of* **must** *by our characterization.*

$P_3 = \mathbf{rec}{:}x.\mathrm{coin}.(\mathrm{coffee}.x \lhd_{20} x \oplus \mathrm{coffee}.x \lhd_{21} x)$
$Q_3 = \mathbf{rec}{:}x.\mathrm{coin}.(\mathrm{coffee}.x \lhd_{20} x)$

Since $L_s(\mathrm{coin}.(\mathrm{coffee}.x \lhd_{20} x)[P_3/x]) = L_s(\mathrm{coin}.(\mathrm{coffee}.P_3 \lhd_{20} P_3)) \trianglelefteq L_s(P_3)$, *by corollary 3* $Q_3 \trianglelefteq P_3$.

Not let $R_3 = \mathrm{coin}.(\mathrm{coffee}.Q_3 \lhd_{20} Q_3 \oplus \mathrm{coffee}.Q_3 \lhd_{21} Q_3)$. *If we can show that* $R_3 \ll^{\mathsf{sym}}_{\mathbf{must}} Q_3$, *then by corollary 5, we can conclude that* $P_3 \ll^{\mathsf{sym}}_{\mathbf{must}} Q_3$. *First since* $Q_3 \trianglelefteq P_3$, *condition (1) in definition 11 is satisfied.*

Here, $M(R_3) = M(Q_3) = \infty$, $S(R_3) = S(Q_3) = \{\mathrm{coin}\}$. *Let* $Q_3' = \mathrm{coffee}.Q_3 \lhd_{20} Q_3$ *and* $Q_3'' = \mathrm{coffee}.Q_3 \lhd_0 Q_3$. *Then,* $Q_3 \overset{\mathrm{coin}_0^\infty}{\Longrightarrow} Q_3'$ *where* $M(Q_3') = 20$. *And* $Q_3' \overset{20}{\rightsquigarrow} Q_3''$ *and* $Q_3' \overset{20}{\rightsquigarrow} Q_3$, $Q_3' \overset{\mathrm{coffee}^2 0}{\Longrightarrow} \circ Q_3$.

Let $R_3' = \mathrm{coffee}.Q_3 \lhd_{20} Q_3 \oplus \mathrm{coffee}.Q_3 \lhd_{21} Q_3$, *then* $R_3 \overset{\mathrm{coin}_0^\infty}{\longmapsto} R_3'$.

We have the following three cases:

(1) For $Q_3 \overset{\mathrm{coin}_0^\infty}{\Longrightarrow} \circ \overset{0}{\rightsquigarrow} Q_3'$, $M(Q_3') = 20 > 0$. *Thus, we check condition (2-b).*

Now $\{0\} \cup \{c | 0 \leq c < 20, R_3 \overset{\mathrm{coin}_0^\infty}{\Longrightarrow} \circ \overset{c}{\rightsquigarrow}\} = \{0\}$. *For* 0, $R_3/\langle \mathrm{coin}_0^\infty, 0 \rangle = \{R_3', \mathrm{coffee}.Q_3 \lhd_{20} Q_3, \mathrm{coffee}.Q_3 \lhd_{21} Q_3\}$. $S(R_3') = \{\mathrm{coffee}\} \subseteq S(Q_3') = \{\mathrm{coffee}\}$ *make the condition hold.*

(2) For $Q_3 \stackrel{\mathrm{coin}_0^\infty}{\Longrightarrow} \circ \stackrel{20}{\rightsquigarrow} Q_3''$, $M(Q_3'') = 0$. *Thus, we check condition (2-a).*
$R_3'/\langle \mathrm{coin}_0^\infty, 20 \rangle = \{\mathrm{coffee}.Q_3 \lhd_0 Q_3, Q_3\}$.
And $S(R_3') = \{coffee\} \subseteq S(\mathrm{coffe}.Q_3 \lhd_0 Q_3) = \{\mathrm{coffee}\}$.

(3) For $Q_3 \stackrel{\mathrm{coin}_0^\infty}{\Longrightarrow} \circ \stackrel{20}{\rightsquigarrow} Q_3$, $M(Q_3) = \infty$. *Thus, we check condition (2-b).* $\{20\} \cup \{c | 20 \leq c < \infty, R_3 \stackrel{\mathrm{coin}_0^\infty}{\Longrightarrow} \circ \stackrel{c}{\rightsquigarrow}\} = \{20, 21\}$. *For* 20, $Q_3 \in R_3'/\langle \mathrm{coin}_0^\infty, 20 \rangle$ *satisfies the condition.*

For 21, $Q_3 \in R_3'/\langle \mathrm{coin}_0^\infty, 21 \rangle$ *also satisfies the condition.*

Henceforth, the transitions form R_3 *are same as those from* Q_3. *Then, we have the condition (2).*

Conversely, for $R_3 \stackrel{\mathrm{coin}_0^\infty}{\Longrightarrow} \circ \stackrel{21}{\rightsquigarrow} \mathrm{coffee}.Q_3 \lhd_0 Q_3$, *there exists no* R *such that* $Q_3 \stackrel{\mathrm{coin}_0^\infty}{\Longrightarrow} \circ \Rightarrow_{21} R$ *and* $\mathrm{coffee} \in S(R)$. *Thus,* $Q_3 \not\ll_{\mathbf{must}}^{\mathrm{sym}} P_3$.

5 Concluding Remarks

In this paper, we have presented a testing framework for real-time communicating processes that is extended to deal with the time constraints for communication. The time domain dealt with is the real numbers. Our calculus is extended by the "timeout" combinator, $\lhd_d$, where $P \lhd_d Q$ is intended to denote the process that behaves like P if P performs any observable event before time d, or behaves like Q otherwise.

We have established the testing preorders for the extended calculus of real-time communicating processes and the alternative characterizations which are fully abstract to the testing preordres, focusing on the "essential" type of timed tests. Moreover, we also have given another type of alternative characterizations, called *symbolic alternative characterizations*, in need to establish a proof technique for the regular class of real-time communicating processes.

Our calculus for real-time communicating processes is based on the sub-calculus of Timed CSP[7] and Timed CCS[23]. We chose such a timeout operator that is presented in[19] as a time-constraint combinator, which is considered general enough. For example, we can define the delay prefix[23] $\epsilon(c).E$ as $\mathbf{nil} \lhd_d E$. In [23], the timeout is presented by the rules that a τ transition must be prior to the time passage. [19] also uses the timeout operation as the time constraint. Since their approach of modeling a process as a graph is also based on a strong bisimulation, it differs from our approach. But, for the untimed processes, the testing preorder can be related by constructing acceptance graphs[6]. It is a future topic to investigate such a relation for the real-time communicating processes.

The "symbolic" technique presented here is studied first in [17] for the strong bisimulation. In [17], a logical characterization is presented as a proof technique. We studied the technique for the testing preorders and established the "symbolic" alternative characterizations as a proof technique. One of the advantages of the testing preorders is that they are formulated as preorders of nondeterminism. In our observation, if the timing to make a choice is uncertain, then it should be treated as a nondeterminism. Thus, in testing preorders, we can conclude the third example in the previous section, while they are treated unequal in the bisimulation semantics.

The symbolic alternative characterizations here cannot deal with the time constraints bound by communications as presented in [4][16]. By using another type of

"symbolic" technique [9], it can be treated in a finite way. This is also a future topic of research.

By excluding divergent processes, the failure semantics[7] for timed CSP characterizes our testing preorders. In our framework, we considered no divergent processes since our main objective is to have some proof technique. Different from the untimed processes, if we have the maximal progress property, a divergent process is considered as a "time-stop" process. Timed CSP deals with this time-stop process. [7] proposes a stronger testing preorders than ours in the sense to distinguish the divergent processes. While, our framework is incapable to distinguish the divergent processes from the inactive processes. We do not know yet whether this lack of distinguishing power is a substantial drawback in modeling the practical real-time programs or not.

For other future work, we need to introduce the composition operator and the expansion theorem. The expansion theorem enabled more process to be converted for our method. And implementing a software tools as the practical verification and debugging method for real-time systems based on the testing preorders is also a future topic of research.

References

[1] J.C.M. Baeten and J.A. Bergstra. "Real time process algebra." *Formal Aspects of Computing*, Vol.3, pp.142–188, 1991.

[2] J.C.M. Baeten and W.P. Weijland. *Process Algebra.* Cambridge Tracts in Computer Science 18. Cambridge University Press, 1990.

[3] T. Bolognesi and S.A. Smolka. "Fundamental Results for the Verification of Observational Equivalence." *Protocol Specification, Testing and Verification, VII*, H.Rudin and C.H.West (editors), pp.165–179, 1987 (North Holland).

[4] Liang Chen. *Timed Processes: Models, Axioms and Decidability.* PhD thesis, The University of Edinburgh, Department of Computer Science, 1993.

[5] R. Cleaveland and A.E. Zwarico. "A theory of testing for real-time." Proc. Logics in Computer Science '91, pp.110–119, 1991.

[6] R. Cleaveland and M. Hennessy. "Testing Equivalence as a Bisimulation Equivalence" *Formal Aspects of Computing*, Vol. 5, pp.1–20, 1993.

[7] J. Davies and S. Schneider. "A brief history of timed CSP." *Theoretical Computer Science*, Vol.138, pp.243–271, 1995.

[8] Hans A. Hansson. *Time and Probability in Formal Design of Distributed Systems.* PhD thesis, Uppsala University, Department of Computer Science, 1991.

[9] M. Hennessy and H. Lin. "Symbolic bisimulations." *Theoretical Computer Science*, Vol.138, pp.353–389, 1995.

[10] M. Hennessy. *Algebraic Theory of Processes.* The MIT Press, 1988.

[11] M. Hennessy. "On timed process algebras: A tutorial." Technical Report Report 2/93, University of Sussex, Computer Science, 1993.

[12] M. Hennessy and H. Lin. "Symbolic bisimulations." *Theoretical Computer Science*, Vol.138, pp.353–389, 1995.

[13] C.A.R. Hoare. *Communicating Sequential Processes.* Prentice-Hall, 1985.

[14] U. Holmer, K. Larsen, and W. Yi. "Deciding properties of regular real timed processes." Proc. CAV '91 (LNCS Vol.575), pp.443–453, 1991.

[15] R. Milner. *Communication and Concurrency.* Prentice-Hall, 1989.

[16] F. Moller and C. Tofts. "A temporal calculus of communicating systems." Proc. CONCUR 90 (LNCS Vol.458), pp.401–415, 1990.

[17] U.Holmer, K.Larsen, and Yi Wang. "Deciding properties of regular real timed processes." Proc. CAV '91 (LNCS Vol.575), pp.443-353, 1991.

[18] R. De Nicola and M.C.B. Hennessy. "Testing equivalences for processes." *Theoretical Computer Science*, Vol.34,pp.83–133, 1983.

[19] X.Nicollin, J.Sifakis and S.Yovine. "From ATP to Timed Graphs and Hybrid Systems." Proc. Real-Time: Theory in Practice (LNCS Vol.600), pp.549–572, 1991.

[20] Tim Regan. *Process Algebra for Timed Systems.* PhD thesis, University of Sussex, Computer Science, 1991.

[21] S. Schneider. "An operational semantics for timed CSP." Technical Report TR-1-91, Programming Research Group, Oxford University, 1991.

[22] Wang Yi. "CCS+Time=an interleaving model for real time systems." Proc. ICALP 91 (LNCS Vol.510), pp.217–228, 1991.

[23] Wang Yi. *A Calculus of Real Time Systems.* PhD thesis, Chalmers University of Technology, Department of Computer Science, 1991.